NOVA IMS

Information Management School

MDSAA

Master Degree Program in

Data Science and Advanced Analytics

AN OLX MOTORS CASE: COMPREHEND USER BEHAVIOUR TO SHAPE RECOMMENDATIONS

Sara Michetti

Internship Report

presented as partial requirement for obtaining the Master Degree Program in Data Science and Advanced Analytics

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

NOVA Information Management School

Instituto Superior de Estatística e Gestão de Informação

Universidade Nova de Lisboa

AN OLX MOTORS CASE: COMPREHEND USER BEHAVIOUR TO SHAPE RECOMMENDATIONS

by

Sara Michetti

Internship report presented as partial requirement for obtaining the Master's degree in Advanced Analytics, with a Specialization in Business Analytics

Supervisor / Co Supervisor: Roberto Henriques

September 2022

STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration. I further declare that I have fully acknowledge the Rules of Conduct and Code of Honor from the NOVA Information Management School.

Sara Michetti

Lisbon, 27th November 2022

DEDICATION

To who supported me along the way.

ABSTRACT

The analysis of online user behaviour can shape the way a business like OLX Motors recommends cars to potential buyers. This paper focuses on different ways of getting insights from the searches and the interactions of the users on the websites. Understanding first how users select filters and how long their searches are, led to a first AB test to check whether suggesting filters to select could or not help the user experience. The results showed that looking at different models in a search could bring users closer to their car of interest. Intuitively, recommending car models would eventually enhance the user's journey. This evolved into utilizing word2vec, a widely used algorithm to explore relationships between words in sentences, to draw similarities between similar cars. The data used is the sequence of users' search sessions and not the technical characteristics of the cars. With this a clustering technique was performed to individualize groups of similar vehicles.

KEYWORDS

Recommender Systems; Machine Learning; Cars Market; User Behaviour; AB Test

INDEX

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

T-sne	T-distributed neighbor embedding
SVD	Singular Value Decomposition
CBOW	Continuous Bag-of-Words Model of word2vec

1. INTRODUCTION

Since the outbreak of Covid-19 in 2020, there has been a surge in buying pressure in the car market. From the restriction of parts for new cars across the globe to the need for a safe way to commute without getting the virus and incomes decreasing (Eric Rosenbaum, 2020; Olivier Hanoulle, 2021). This, but not only led people to start looking more into the used cars market (Olivier Hanoulle, 2021). Additionally, the pandemic led to an accelerated run to digitalization. Users across platforms are now expecting easiness and personalization while using them, especially when it comes to using mobile phones and non-desktop devices. It is estimated that used-car buyers spend 40 percent more time researching online than new-car buyers (Ben Ellencweig, 2019).

In this spirit, the OLX group has been working on finding solutions to facilitate the usage of the websites of the classified business, which involves listing used cars. The increased supply and demand led the business to look more closely at the needs of the users in search of a car. When facing the platforms, users need to deal with two essential but also intricated components of a search: the filter selection (and their options) and the relevance of the cars shown to them. By relevance we mean how relevant is the result of the search conducted, which includes also the recommendations.

Through some research and user questionnaires conducted in 2021 by OLX Motors, it was clear that for buyers new to the used car market, it is not easy to understand the characteristics of the car they are looking for. In fact, the first data analysis found that 12% of the users started their search with no filter selection. Usually, people buy a car because of a change in their lives: a young person who starts a new career and needs a car to commute, a newborn arriving to the family, or the need to sell the previous car because it is too consumed. These are just a few examples of users we get to analyze. Not only is the knowledge of the users not complete, but the website browsing experience can be tedious and lead users not to find what they are looking for. Platforms must have ways of leading the customers, suggesting appropriate items, and filters by an easy and non-obstructive UX design.

This thesis is done on the most popular platform in the OLX group for used cars otomoto.pl (https://www.otomoto.pl), in the Polish market, which holds more than 150000 active listings at any given moment. This research started with a clear business need: help the users in their search journey, in particular in the filter selection phase. From this starting point, various analysis and explorations were conducted. The aim was to first understand how users search for cars, interact with the filters, how successful are their searches and how long does it take them to find a car they are interested in. Consequently, after a first iteration of a baseline model suggesting to the user the next possible filter to select, new explorations were performed thanks to the insights from the AB test. Finally, the aim shifted to suggesting cars, developing a machine learning based model with word2vec, an algorithm that transforms words into vectors and understand the similarities between them through neural network. This outlines how users interact with different car types in conjunction with the k-means clustering technique.

Until now, different recommender systems have been developed: content-based, collaborative filtering, and hybrid options, including session-based recommendations. Content-based was researched, for example, by (Wang et al., 2018). In their paper, they propose a model to recommend computer science publications based on the context of the publications (words, frequency etc.). Collaborative filtering discovers similarities between items and users based on attributes from both. Instead, the session-based system (Esmeli et al., 2020) best suits this research's needs: identifying similar items after checking the users' interactions with the items in specific search sessions.

Not only can session-based recommenders be calculated via word2vec, an algorithm that learns word embeddings using shallow neural network, but also recurrent neural network can help with the task. In (Zhou et al., 2018), the authors went through different u sers interactions on an e-commerce platform. They managed to discriminate between micro behaviors and time variables to assess the similarities between items. This means that not only does the relationship of the items clicked by users matter, but also the type of interaction performed and for how long the users were checking out a particular item.

Aside from using similarity metrics to assess the recommendations of word2vec, the T-distributed neighbor embedding (t-sne) can be used to understand the closeness between multiple items. This is especially useful when the goal is not only to recommend but to understand which kind of different clusters and behaviors are available in the data analyzed. T-sne is a dimensionality reduction technique that retains the local structure of data and that helps to visualize large datasets with high dimensionality with a contained use of computer resources. Vectors that are similar in a high-dimensional vector space get reduced their dimensionality to two or three dimensions which allows it to be plotted and visualized (van der Maaten & Hinton, 2008).

What appears to be missing in the literature is the relationship between the vector embeddings of word2vec and the visualization of them via t-sne, especially when it comes to identifying different types of clusters based on the interactions of the users with the items. Only in the paper focused on difference and similarities between dialogues of Korean and Japanese speakers, (Cho & Yoon, 2017) raised the problem of the difference between hyperparameters in t-sne and how they affected the way the datapoints in their plots were distributed.

The goal of the research is to use embeddings algorithms, namely word2vec, to understand the relationship between items with a session based perspective, and with an additional aim of finding different types of behavioral clusters that can discriminate between users and cars they are looking for. This is done for not only data science teams to understand, but also for the broader business which can support and understand better the way we work in data science with visualizations and more crisp information. The data used for this purpose is not only what is described by (Esmeli et al., 2020), but it wants to take into consideration different types of users' interactions without the need of discriminating between them like (Zhou et al., 2018) did.

2. LITERATURE REVIEW

Recommender systems existed before the e-commerce revolution that took place with the invention of first web browser in 1990. Considering that the first secure internet shopping transaction was performed in August 1994 recommendations started in the 1970s, with the computer librarian Grundy, a solution created by (Sándor Apáthy, 2021). The system relied on users' preferences, collected through questionnaires, to recommend books to people in the same group. A first pitfall was spotted in this system: interviewers were answering the questionnaire not based on their preferences but on what they thought would make them look better (for example reading classic literature).

Later, collaborative filtering started being used. This system looks merely at the users' preferences and tries to recommend objects based on similar user. The first of these recommendations was a document recommender by Xerox PARC. Users could like or dislike a document; hence, the system could rank thematic documents based on relevance (Sándor Apáthy, 2021). Not long after, more businesses started using content-based filtering in the music and cinema fields.

Content-based filtering is born as an information retrieval technique that considers two main dimensions: the characteristics of what is being recommended (for a movie, it can be the genre, the release year, the director etc.) with the user's preferences for these characteristics. Every time a user watched a movie on a streaming platform or rated a show, this information would be linked to the person's profile. The first time a similar approach was used was before e-commerce, in the 1960s, by the University of Cornwall: they tried to automatize the indexing of documents. This is the closest form of recommendation that we know today: two similar documents are the ones with the smallest angle between the two vectors that describe them. Every text was imputed into a vector with its classified characteristics. A more recent discovery was made by the Music Genome Project in 1999: they collected 450 songs properties, and these were used to recommend songs to users with similar music preferences. The pitfall of content-based filtering, though, is the blind corner problem. In this case, a user listening to something very specific, like horror movie soundtracks, would probably get recommended just this type of music genre, creating a blind spot in their playlists (Sándor Apáthy, 2021).

To obviate this problem, researchers started looking into the combinations of the two systems described before. It was already looked into by Stanford university in 1994. This technique is widely used today by Netflix, and Amazon, among others (Sándor Apáthy, 2021). The latter recommends products based on a cooperative filtering technique. Not only they checked the users' preferences, the characteristics of the products, and similar users' behaviors, but they also considered the history of the browsing experience of the same user (Linden et al., 2003)

As it is prohibitive to inspect all of the users' purchase histories online and offline, Amazon found a different way of calculating how related items are to each other (Larry Hardesty, 2019). This relatedness metric was based on differential probabilities and is described by Larry Hardesty as *item B is related to item A if purchasers of A are more likely to buy B than the average Amazon customer is. The greater the difference in probability, the greater the items's relatedness*. This metric was first presented in the research conducted by Amazon (Linden et al., 2003). Since then, several steps have been taken to better the recommendations, especially in the Prime Video business. The goal was to solve the matrix completion problem: users on the rows and movies/shows on the columns. If the user watched a show, the cell would contain a 1, if not it would be blank. The goal of completing the matrix is to fill it with the probability of a customer seeing any show or movie. For this, they started using deep learning and autoencoders. These learn to return the same input data after going through bottleneck layers and nodes (Larry Hardesty, 2019). They managed to increase performance compared to collaborative filtering when they sorted the history of viewings chronologically so that they could predict the viewings for the future and not only similar items (Larry Hardesty, 2019).

A different way of computing recommendations for e-commerce platforms can be found in the session-based system. In their research, (Esmeli et al., 2020), were able to identify similar items after checking the interactions of the users with the items in specific search sessions. Additionally, they calculated also the similarity between the items and imputed these metrics in the models. They showed that considering the pattern of the interacted items similarities in the user session can lead to better determination of the intention of possible next purchase.

Not only session-based recommenders can be calculated via word2vec, algorithm that learns word embeddings using shallow neural network, but also recurrent neural network can help with the task: in their paper, (Zhou et al., 2018) went through different users interactions on an e-commerce platform and managed to discriminate between different so called micro behaviors and time variables to assess the similarities between items; this means that not only relationship of the items matters if users clicked of them, but also which type of interaction was performed and for how long the users were checking out a certain item.

The main technique for natural language processing used in this thesis is word2vec. The next paragraphs are dedicated to it.

1. WORD2VEC

The unsupervised algorithm word2vec is a method that tries to map words and sentences into a lower dimensional vector space. This space will capture the relationship that words share semantically (Barkan & Koenigstein, 2016a). Word2vec is also recognized as Skip-gram with Negative Sampling (SGNS), which has been extensively proven to be outstandingly useful in both NLP tasks and other applications (Barkan & Koenigstein, 2016a).

The main problem that (Mikolov et al., 2013a) tried to solve is the high computational cost of using words as atomic units instead of vectors. Using words as atomic units did help in understanding the similarity between words and (Mikolov et al., 2013a) switched to a vectorial perspective. Their goal, that they reached with the definition of Continuous Bag-of-Words Model (CBOW) and Continuous Skip-gram models, was not only to understand which words are mostly similar to each other, but also to calculate the degree of similarity they share. In their first paper they want to try to maximize accuracy of these vector operations by developing new model architectures that preserve the linear regularities among words.

Regarding the CBOW model, the goal was to build a neural network that does not take into consideration the order of the words in the sentence when computing the projection (Mikolov et al., 2013b). Having *N* previous words and consecutive, *N x D* the dimensionality of the projection layer, *W* the size of the vocabulary, the training complexity's (reference) goal is to classify the current word.

$$Q \ = \ N \cdot D + D \cdot \log_2(W)$$

(1)

Instead, the Continuous Skip-gram model tries to predict the words within a predefined range (*C* in the formula (2)) before and after the word taken in consideration now. The further away the words, the less weight the words will get. The training complexity for this model is (Mikolov et al. 2013):

$$Q = C \times (D + D \times \log_2(V)).$$

(2)

The goal is to maximize the following term:

$$\frac{1}{K}\sum_{i=1}^{K}\sum_{-c\leq c,\, j\neq 0}\log p\left(w_{i+j}\ \ |w_i|\right)$$

(3)

Where $p(w_j \cap w_i)$ is the SoftMax function:

$$p(w_j \cap w_i) = \frac{\exp(u_i^T v_j)}{\sum_{k \in I_W} \exp(u_i^T v_k)}$$

(4)

This SoftMax function is computationally expensive, therefore it was replaced by a negative sampling technique (Mikolov et al., 2013a). The idea was to have a model that could differentiate data from any type of noise via logistic regression always retaining the quality of the vector. The $p(w_{t+j} \cap w_t)$ was replaced by the expression (5) where k are the negative samples for every sample of data:

$$\sigma(u_i^T v_j) \prod^{N}_{k=1} \sigma(-u_i^T v_k)$$

(5)

The value of k , Mikolov et al (2013) say, can be between 2 to 5 for large datasets and can get to 20 in case of small training datasets.

A common problem that they had to face is the difference in frequency between popular and unpopular words. A clear example is between prepositions, articles that do not enclose a real meaning on the contrary of more meaningful words, that probably are less present in texts. To remedy this challenge, they used a subsampling approach. This means that each word can be discarded based on a probability that takes into consideration the frequency of the word itself $f(w_i)$ and a threshold t that empirically was set at 10^{-5}.

$$P(w_i) = 1 - \sqrt{\left(\frac{t}{f(w_i)}\right)}$$

(6)

Like this, words with frequency greater than t, would be subsampled while maintaining the frequencies of the words. This threshold was proven to be significant enough to improve the accuracy and the speed of learning for the model (Mikolov et al., 2013a).

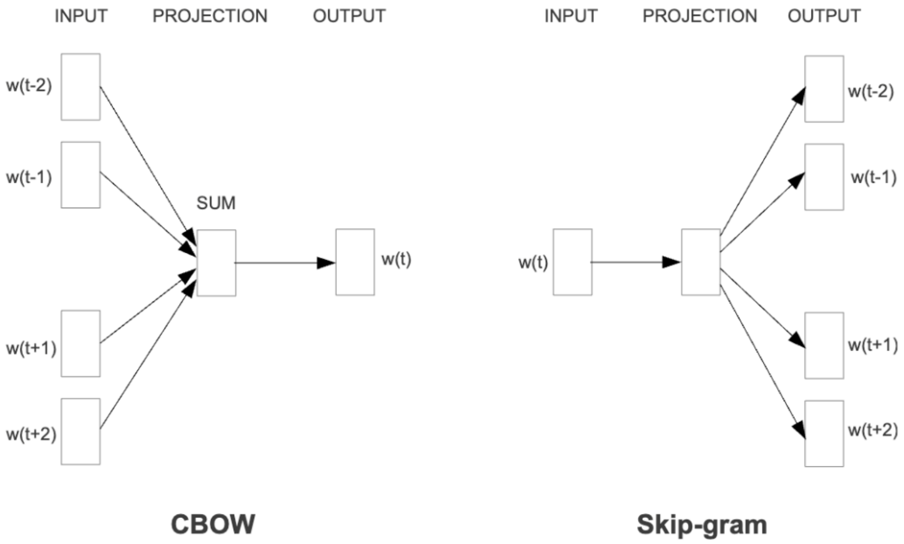


Figure 1 - Difference between CBOW and Skip-gram in word2vec

As these models are based on words, researchers then started looking not only into sentences, if not into sequences of items, specifically items searched, bought, clicked by users on websites. This technique is called item2vec. This was first researched by (Barkan & Koenigstein, 2016b) which uses the idea of Collaborative Filtering and word2vec to create embeddings for items in a latent space. This is helpful for what concerns the cold start problem, just to name one, because it does not rely on user data but only on the relationship between items. This approach consistently took over the basic Singular Value Decomposition (SVD), especially when the data points considered are further away from the most popular ones (Table 2, Barkan & Koenigstein, 2016b, p. 4) and because of the subsample technique cited before.

From this moment onwards various research has been made in both collaborative filtering and content based methods: the increasing amount of data regarding users' activity and the need to combine both items information with users' interactions, led research to explore the hybrid methods. For example, the graph-based hybrid recommender system tackled by (Chow et al., 2014) which is a personalized page rank. It uses as input both the user and feature correlation graph to forecast the user preferences; they designed this in for a ranking problem, to suggest similar games in mobile applications. Another example can the (Wang et al., 2018), in which the goal was to take in consideration and discriminate between different users' interactions, called by them micro behaviors, on the websites and use them as inputs for their model so called Recommendations of Micro Behaviors. They added on top of the embeddings layer a recurrent neural network layer to capture the sequential information of micro behaviors.

2. METHODOLOGY

For this report a specific methodology was used, which shares some characteristics with the CRISP methodology (Cross-Industry Standard Process of Data). This one allows for flexibility in those cases where the study needs to start from data and business understanding to create a model and evaluate it and possibly deployment. Being a non-rigid methodology, it allows, for example, to go back to the business understanding once a model evaluation is done. The process can look like the one on the left in Figure 2. .

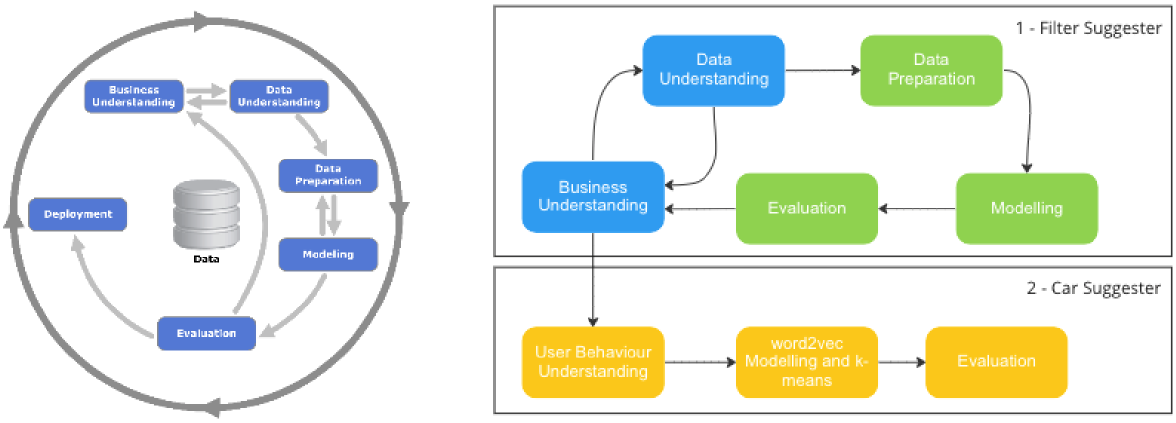


Figure 2 – Differences between CRISP and current methodology

Instead of the full CRISP cycle, this research has two main phases (right diagram in Figure 2.): the filter suggester and the car suggester one.

In the first part the goal was for the business to understand how users perform searches and especially, how do they behave in the filter selection phase of their search. After this first data understanding, the first model created was a baseline that had as objective to suggest to the user the next possible filter to select. This model is based on the most frequent filter selection and does not involve machine learning. After testing it with the users via AB test, it was essential to go back to the root question and analyze the problem again with new insights from the first experiment. Like this the car suggester phase started. The learnings from the first stage helped to look at the problem from a new perspective and create a new solution with a machine learning model that uses the word2vec algorithm. Instead of suggesting the next possible filter to select, the car suggester aims to suggest similar cars to users that are in the listing page of the website. To group together different user behaviors, k-means was employed.

1. DATA COLLECTION AND EXPLANATION

The data used for this thesis comes directly from the databases of the company. Tracking events are stored in different tables based on different dimensions. In this case the data is taken from tables that take into consideration the actions of the users (especially the users looking for a car) on the website. All the analysis and the algorithms mentioned are run in Python and for querying the data MySQL language was used. To do the latter one, aws Athena and S3 were the most employed. The data extraction is usually done via a jupyter notebook: both aws credentials and profiles are needed to accomplish this and after running the query, the data would get saved on the local machine under a specific folder, usually called with the day of when the data was extracted.

Before starting collecting the data some effort was put into understanding which are the users' interactions that get saved and how are these defined. In the next sub chapter, the most important definitions are explained. These terms are fundamental for the understanding of the analysis conducted and the results of the user behaviour the otomoto website outlined in the next chapters.

Listing page and ad page

The listing page is the place where it is possible to see all the cars on sale one below each other. While the ad page of a car, is a specific web page in which the car that is trying to get sold is displayed with all its characteristics, including the price, pictures and the information of the seller.

Events

Tracking events can vary, and they are continuously collected and inserted into the final tables. An event can be the opening of a specific car ad, scrolling the page, selecting specific filters or refreshing the page. What is of most interest in this session are all those events that take into consideration whether a user might be interested in a specific car or type of car. For example, when a user is browsing a car ad page and goes to check the map of the seller, or a user that checks the phone number of the seller or adds a specific ad to the favorites.

There is also another range of actions that are not taken into account in this session. These are also, but not only, the action of clicking on an ad of a car, or just scrolling through the ad page without significantly generating successful events, or as they are also called: meaningful interactions.

Another event that we account for in the analyses is the listing one. A listing event is simply the action of a user when they apply new filters on the main listing page.

Successful and unsuccessful events

To understand these concepts, it can be useful to introduce the definition of a successful event. Contrary to a listing event, a successful event is an action that leads the user closer to a car they might like, and it is an action that is accomplished in the page of a specific car and not on the listing page. In OLX Motors we consider a successful event anything that ranges from trying to contact the seller, to adding the car to the favorites, or checking where the seller is precisely. In total we account for 11 different successful events, or also called meaningful interactions.

Having understood the definition of a successful event, it is now possible to understand the difference between a successful search and a non-successful one. The first one is, by definition, a search that leads to a meaningful interaction passing by a listing event, while the latter ones are searches that most of the time go through listing events and car views without any real interesting interaction.
Throughout this report the goal is to understand how users interact meaningfully with the website, and also grasp the concept of similar searches and similar cars thanks to the concept of successful searches.

For analysis purposes the dataset extracted was taking into consideration the filters applied by the users in the listing page of the website, what was defined before as listing event. The number of filters taken into consideration for these analyses is 23. They range from car model, car make (the brand of the vehicle), fuel type, the range of years of production, mileage and so on.

The dataset used for the first baseline is retrieved from one table that collects all the user interactions on the website in any given page. Each row of the dataset, considering the limit of rows is normally set at one million, is an event, which is an action of the user. For the baseline frequency model (explained in the next subchapter), called filter suggester, only listing events were taken into consideration. In each row of the dataset three main data points are used: the user (called session_long in the tables), the event (listing in this case), and the filters (if applied by the user, the field is populated with the input, if not it remains empty). A series of listing events from a user might look something like Table 1.

session_long	event	server_date	make	model	from_year	from_mileage
2148920017	listing	2022-07-01 18.15	bmw	NaN	2017	100000
2148920017	listing	2022-07-01 18.16	audi	a3	NaN	100000

Table 1 – Example of listing events table

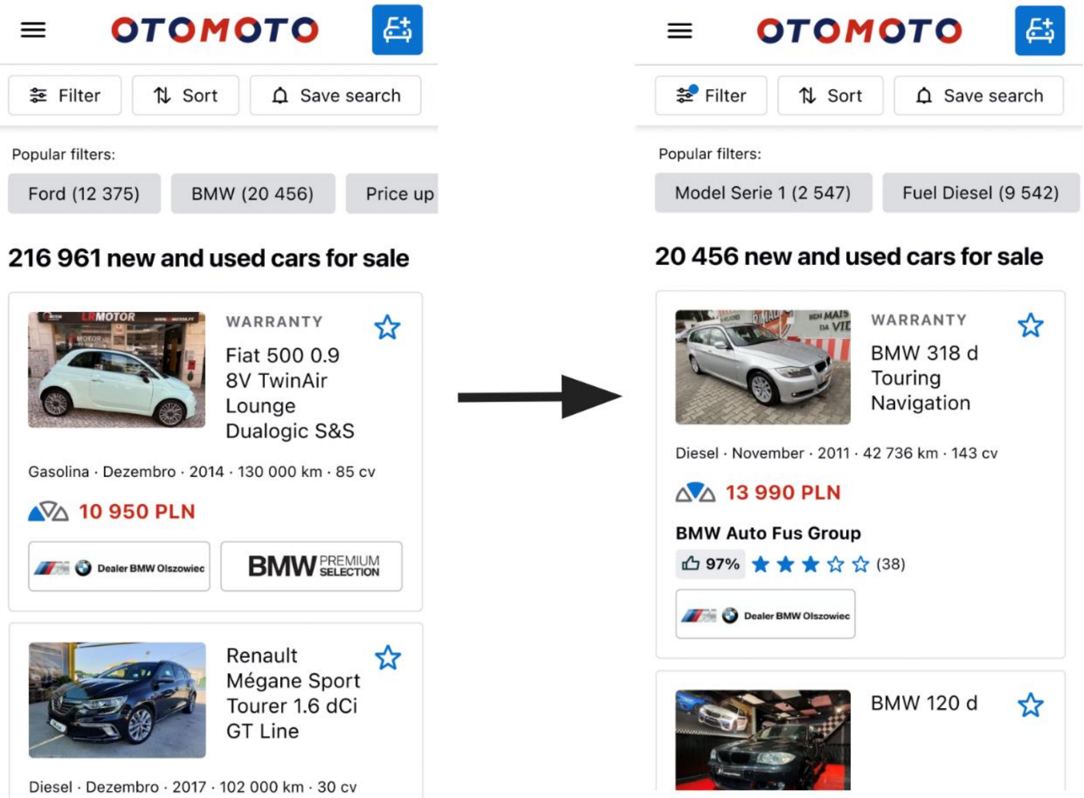
In this case the user filtered the webpage firstly by cars with the *BMW* make, which production year goes from 2017 and the mileage of the cars starts at 100000 km, and then, after one minute at 18.16, they filtered by *Audi A3* cars, without a production year filter, but they still wanted to look at cars from 100000 kms. This is a sample of the table, as it was mentioned before there are 23 different car characteristics filters taken into consideration.

2. Frequency model

All these analyses, which results are outlined in the following chapter, helped to create the first baseline model that was built by the team: for every possible combination of filters, the model returns five possible different filters to select along with the ones already selected. This is because the goal was to give recommendations to users regarding the next filter to select on the listing page.

The final cleaned dataset takes into consideration all the possible filter combinations that the user applied, while dropping if there are any duplicated combinations for the same user session.

As an example, in Figure 3, considering that all the experiments that the team are running are on web mobile, on the left screen the users did not select any filters and just accessed the listing page, and the *popular filters* are the outcome of the model when no filters are selected. If then the user decided to click on the filter *BMW,* then the second screen would appear with new recommendations.



216 961 new and used cars for sale

WARRANTY
Fiat 500 0.9 8V TwinAir Lounge Dualogic S&S
Gasolina · Dezembro · 2014 · 130 000 km · 85 cv
10 950 PLN
Dealer BMW Olszowiec
BMW PREMIUM SELECTION

Renault Mégane Sport Tourer 1.6 dCi GT Line
Diesel · Dezembro · 2017 · 102 000 km · 30 cv

20 456 new and used cars for sale

WARRANTY
BMW 318 d Touring Navigation
Diesel · November · 2011 · 42 736 km · 143 cv
13 990 PLN
BMW Auto Fus Group
97% ★★★☆☆ (38)
Dealer BMW Olszowiec

BMW 120 d

Figure 3 – Filter suggester example

AB TEST AND METRICS

For the test on the frequency model, we relied mostly on metrics which track the user interaction with the website when they are exposed to the filter suggester. The experiment was taking into consideration only mobile users on the webpage (not on the apps) with 50% of the traffic redirected to the control group and the remaining 50% redirected to the experiment group, the filter suggester one. Once a user is assigned to a certain group, A or B, the assignment will stick until the end of the experiment.

The hypothesis that was going to be tested is the following: *"If we suggest the next filter to select, then the user would have more and faster meaningful interactions because they will* *have a more guided search session*." The metrics used are outlined in Table 2 – Metrics for the filter suggester experiment below:

Metric	**Description**	**Where is calculated**
Listings per users with meaningful interaction (primary metric)	number of listing events for users that reached any meaningful interaction.	Online
Time to success	how many minutes it takes to get from the first listing to the first successful event	Offline
Engagement time	how many minutes does a search session last	Offline
% users with meaningful interactions	users that reached a meaningful interaction / total users	Online
% of searches with meaningful interactions	how many searches have meaningful interaction on the total number of searches	Offline
Meaningful interactions per user	number of meaningful interactions per user	Online
Meaningful interactions per listing	number of meaningful interactions per listing event	Online
Listing per user	number of listing events per user	Online

Replies per listing	number of replies events per listing event	Online
Replies per user with listing	number of replies events per user with listing event	Online

Table 2 – Metrics for the filter suggester experiment

These metrics calculated both online and offline depending on the availability of the data, were crucial to understanding the performance of the filter suggester.

The experiment was set on a platform created by and for OLX. Once the experiment is set up with the metrics and the groups, the backend and frontend team can connect to that specific experiment so that all the assignments are correctly made to the users. In this tool, the metrics get different statistical tests based on the distribution of the data of the control group and the variant group.

3. WORD2VEC ALGORITHM AND K-MEANS

To finish up the analysis on user behaviour also word2vec was employed. This kind of algorithm helps to understand how similar cars can be between each other based, not on their characteristics, but on the interactions of the users with each of them. For this purpose, the dataset was brought down to meaningful interactions only, meaning that the listing events were not considered for this analysis. This is because the goal was to understand how users are interested in similar cars. As explained before listing events can have little importance on how involved a user is with that specific car model. For every user the dataset would return only the car models they had a meaningful event with, timely ordered.

The dataset used to train the word2vec algorithm differs from the one of the baseline frequency model because the events taken in consideration are meaningful interactions and not listing. Also, in this case, the focus is on the ads interacted and not on the filters applied.

This means that the dataset is a sequence of user sessions, and for each one of them the make and the model of the ad interacted with is specified. Being the goal to see the relationship between cars seen in a user session and the quantity of cars the user sees, all the duplicated data per user were dropped; additionally, to gather more sessions with two or less make model combination, were dropped.

session_long	event	server_date	make	model
2148920017	meaningful interaction	2022-07-01 18.15	bmw	Serie-1
2148920017	meaningful interaction	2022-07-01 18.16	audi	a3
2148920017	meaningful interaction	2022-07-01 18.16	audi	a4

Table 3 – Example of input dataset for car suggester

For this specific analysis the Continuous Bag-of-Words (sg parameter in Table 4 – Word2vec hyperparameters) model was used from the Gensim library. As previously explained it tries to predict the current word based on a predefined number of words before and after the one currently taken into consideration. To check the most similar car it is possible to just compute a similarity metric to return the closet vector in the embedding space (similar_by_vector function of the word2vec). The hyperparameters applied can be seen below in Table 4 – Word2vec hyperparameters.

Word2vec Parameters

window = 3
vector_size = 25
epochs = 25
sg = 0
hs = 0
negative = 15
alpha = 0.05
min_alpha = 0.0006
seed = 14

Table 4 – Word2vec hyperparameters

To get to this hyperparameter space a grid search was performed. As the Gensim library does not support this kind of exploration, the team built an ad-hoc grid search for this purpose. This generates one vector per word with 25 dimensions. The “hs” parameter in conjunction with the “negative” one, states if the negative sampling will be use and with how many words, fifteen in this case. The negative sampling is used instead of computing the SoftMax function on the whole vocabulary.

After training the model, in order to assess some meaningful clusters, k-means algorithm was deployed. To look for the ideal number of clusters both hierarchical clustering and the "Elbow method" were used. The R-squared was used to understand which distance method was most suitable and then the result of the dendrogram was used against the elbow method graph.

In order to then visualize and get a better understanding of the embedding space, T-distributed neighbor embedding (t-sne from the library sklearn) was applied. T-sne helped to go from a 25-dimensional space to a 2-dimensional space and infer some behaviors via visualizations also thanks to metadata (more on this in the results chapter). Regarding the scatterplots the library used is Bokeh, a Python library for creating interactive visualizations.

3. RESULTS AND DISCUSSION

In the first part of this chapter various analyses are laid down to better understand the context in which the work was done. Thanks to these findings it was decided to continue the work and test the first frequency model for the filter suggester. The results of the AB test lie in the Evaluation AB Test subchapter too.

In the second part it is explained the outline to change the focus from a filter suggestion goal to a car suggestion goal. The focus is on which analysis helped to draw those conclusions helped by the unsupervised algorithm of word2vec combined with the clustering technique of k-means.

1. First Business and Data Exploration

In order to understand users' behavior, an in-depth analysis of the business and the problem was put into place. Starting from a raw understanding of why users buy used cars and which are their needs, the focus then shifted on how the websites work, how they are structured and in which way the users perform searches.

1. Filters analysis

In this phase multiple data analysis have been run and the key takeaways can be summarized in the following paragraph (Figure 4, Figure 5, Figure 6):

- *make* and *model* are the filters used the most (more than 70% of the searches apply them)
- *from year* filter is used 32% of the time
- 6% of the searches happen without selecting any filters
- the rest of the filters are used less than 15% of the time per search

- the least used filters are the size of the motor, what is the minimum mileage for the cars, the car color, other equipment and the power of the engine.

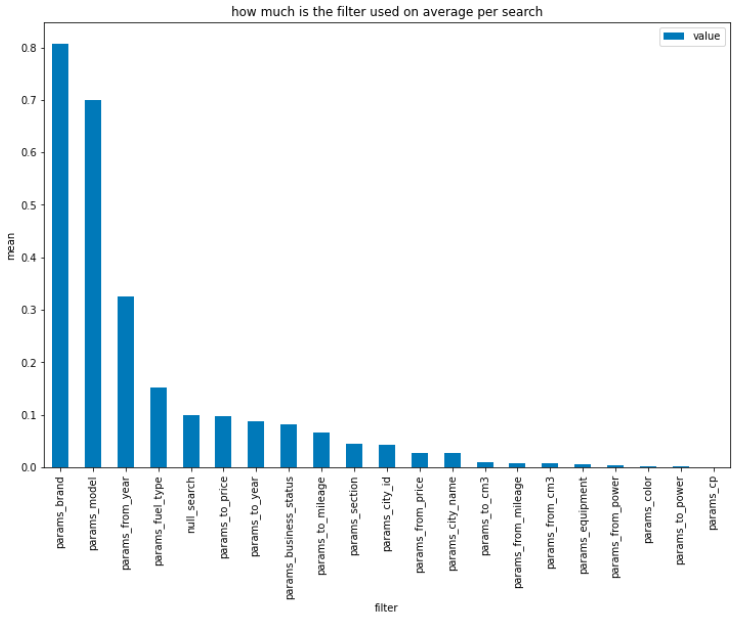


Figure 4 – Average filter usage per search in percentage

Additionally, as it can be seen below in, users mostly use two filters per search, with the combination mostly chosen being *make* and *model* which account for 23% of the searches (Figure 4). 90% of the time users use from 1 to 4 filters. This was already suggesting that users tend to use less filters, especially in their first part of the search. Also, applying a copious number of filters can lead to zero results pages.

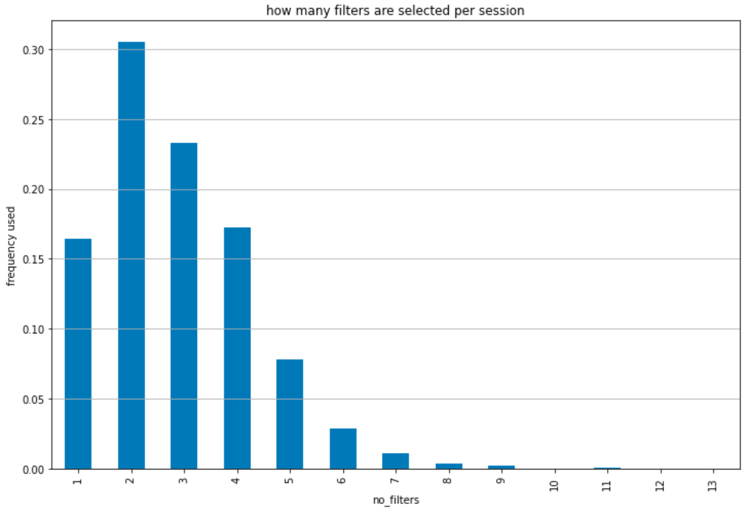


Figure 5 – Number of filters selected per session

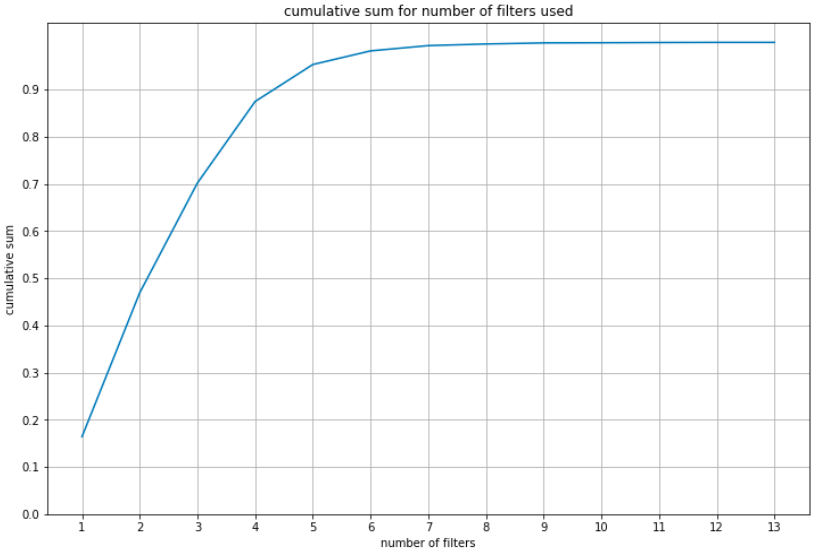


Figure 6 – Cumulative sum of filters used per session

Lastly, the top ten combinations all include brand or model (or both) and it accounts for 61% of the total searches (Figure 7).

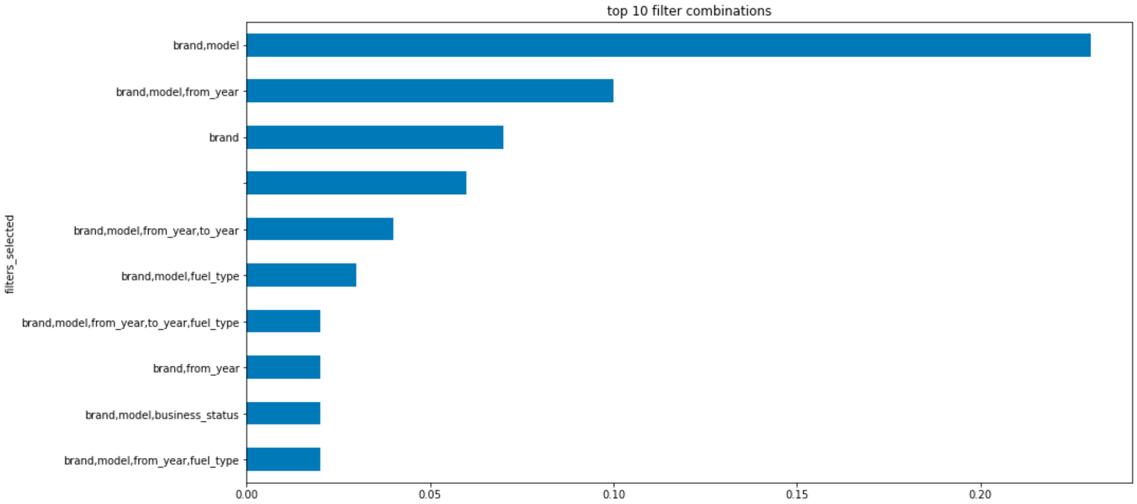


Figure 7 – Top 10 filter combinations

2. Time and definition of success

Not only was it needed to check how users perform their searches but also how long it takes them to carry them out. To do so, searches were split in two different categories: searches that were successful and searches that were unsuccessful.

Different type of searches were identified:

1. non successful searches: 65%
2. Searches that started with a listing and landed to a successful event: 20%
3. searches with no listing events, but just meaningful interactions: 12%
4. success to listing searches: 2% (these rows were dropped)

The time analysis conducted revealed that on average searches that start with a listing event and end with a successful event take 15.9 minutes, while for the median it gets down to 6.8 minutes. What is interesting is to check the last percentiles of the distribution: users that lie on the 90th percentile take more than 35 minutes to reach the first successful event (second table) and more than 32 minutes to reach the last successful event.

search type	min	median	mean	80% percentile	90% percentile	max	std
From listing to success	0.03	6.80	15.97	21.68	35.74	6946	99.52
Only listing searches	0.00	0.08	12.80	21.95	43.65	301.52	24.70

Table 5 – Statistics of total time and time to success for different type of searches in minutes

search type	min	median	mean	80% percentile	90% percentile	max	std
From listing to success	0.03	6.80	14.90	19.87	32.97	6946	99.33

Table 6 – Time to reach a successful event in minutes

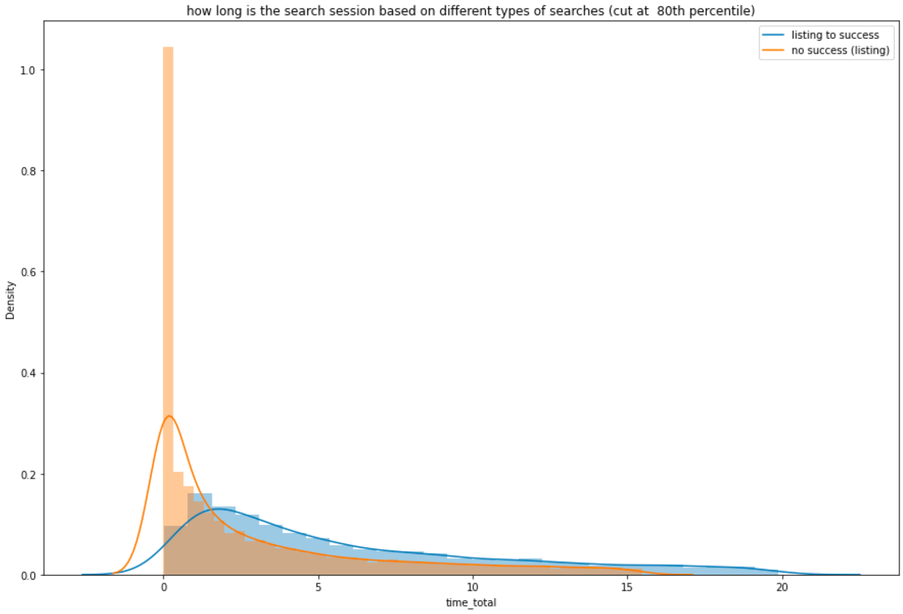


Figure 8 – Distribution of time per type of search

Non-successful searches, that are the ones that do not ever get to a meaningful interaction, have a lower time median, 2.4 minutes, but the mean and the 90th percentile are similar to the listing-to-success searches, even if a bit lower.

These findings together helped the team to understand that not only is it crucial for users to apply the correct filters, but also that it is important to bring down the time per search of the users. Hence it was decided to continue with the first baseline model, a frequency based one.

1. Evaluation: AB Test

AB testing in OLX is done via a custom tool which allows the teams to set the metrics that we want to compare between the control group and the *filter suggester* group.

The AB test was run for 13 days in February 2022 on the Otomoto platform for mobile users only.

Laqesis Metric	**Result**	**Statistical Significance**	**Comments**
Listing per users with meaningful interaction	-4.84%	Under Minimal Sample Required	Since the sample was unreachable, it can't be used for conclusions. But the direction of it is the one we were hoping for.
Listing per user	-0.78%	Under Minimal Sample Required	
Replies per listing	+1.86%	**Positive**	Resulted in more replies on listing events (formula: replies / listing events)
Meaningful interactions per user	+1.71%	Under Minimal Sample Required	
Meaningful interactions per listing	+3%	Under Minimal Sample Required	

% users with meaningful interactions	+2.41%	Under Minimal Sample Required	
Replies per user with listing	-0.29%	Under Minimal Sample Required	
Repliers per all users	-0.69%	Under Minimal Sample Required	
Repliers per all users	-1.23%	Under Minimal Sample Required	

Table 7 – Results of filter suggester experiment

The most interesting results are summarized below:

- 1.49% of the users clicked on the suggested filters;
- The filters suggested with higher ratio of clicks/shows, are the make filter, followed by the model filter. Clicks/shows is the ratio between showing the suggestion of a particular filter to the user (for example the model of the car) and if the filter was used or not;
- The most clicked makes were *Mercedes-Benz, Volkswagen, Audi and BMW* all with a ratio higher than 4%;
- The contacts to the sellers increased significantly by 1.86% compared to the control group;
- The rest of the metrics set did not reach any statistical significance because of a low number of data points collected.

Regarding the minimum sample size, the custom platform suffered from some tracking and assignment problems hence most of the metrics could not reach any statistical significance.

2. User Behavior analysis via Word2vec and K-means

Thanks to the AB test and other offline analysis, it was clear that users still prefer selecting only make and model instead of getting deeper into the filter selection, especially when in their first approach to the search. What was interesting to find was that users that have the most meaningful interactions are the ones that, throughout their searches, had various listing events with different makes or models.

If we split the searches by unsuccessful, successful and searches with just reply events, we see different patterns. The number of brands selected by users which searches led to a reply event and successful events is significantly higher than the searches without a successful event (Figure 9). In fact, for reply and successful searches we see that 2,3,4 brands account for respectively 60% and 35%, while in the non-successful searches we see the 0,1 and 2 prevailing with 79% of the total searches. And, even more interesting, when only one brand is selected, reply searches and successful ones account respectively only for 6% and 4.5% of the total, while for the non-successful searches this ratio increases to 54% (and if also "no brands" would be taken in consideration, the percentage would be 67.5%).

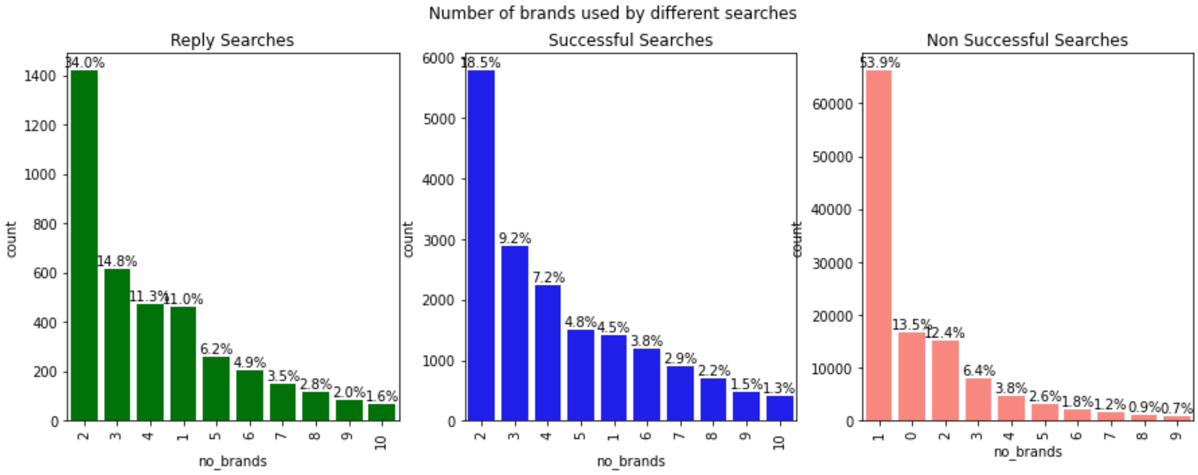


Figure 9 – Number of brands selected by different searches

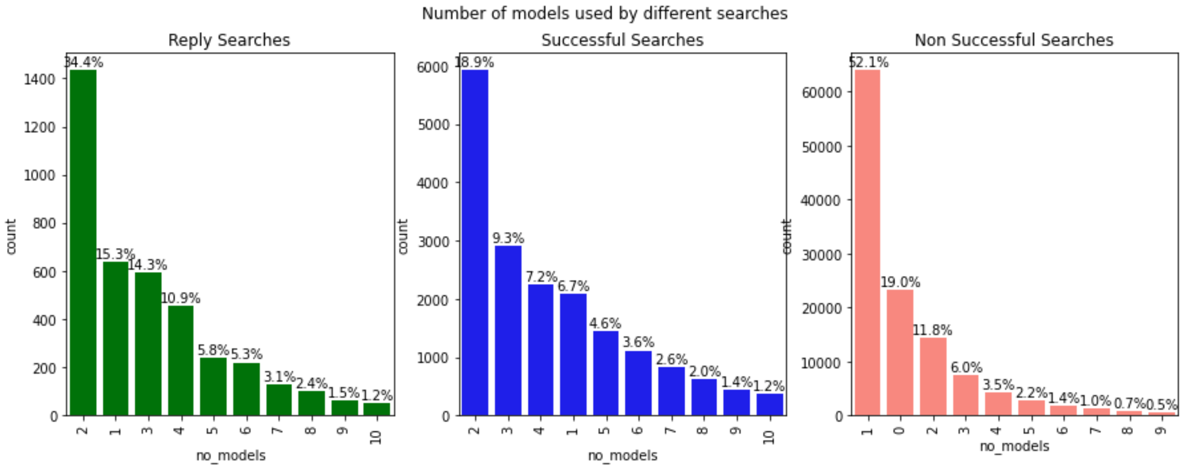


Figure 10 - Number of models selected by different searches

A similar conclusion can be taken looking at the number of models selected by the different search types (Figure 10). Focusing on the non-successful searches it is possible to see that most users use one or no models at all.

The conclusions that can be drawn from this and the AB test results is that probably the best way to help users succeed in their searches is to make sure they see different car models in their search, but of course they should all relate to each other. Meaning that they should be similar cars, similar objects. In this moment the project moved from a filter focus to a car focus, and it got renamed to car suggester.

1. Word2vec and K-means

As previously explained in the methodology chapter, word2vec can greatly help understanding the relationship between searches and car models. As word2vec is unsupervised, one of the ways to assess the results is by visualizing them via scatterplots of t-sne.

In these scatterplots every point is a model of a car and in this specific plot (Figure 11) every circle is colored by its make. It is interesting to notice how cars of the same brand are distributed fairly close to each other.

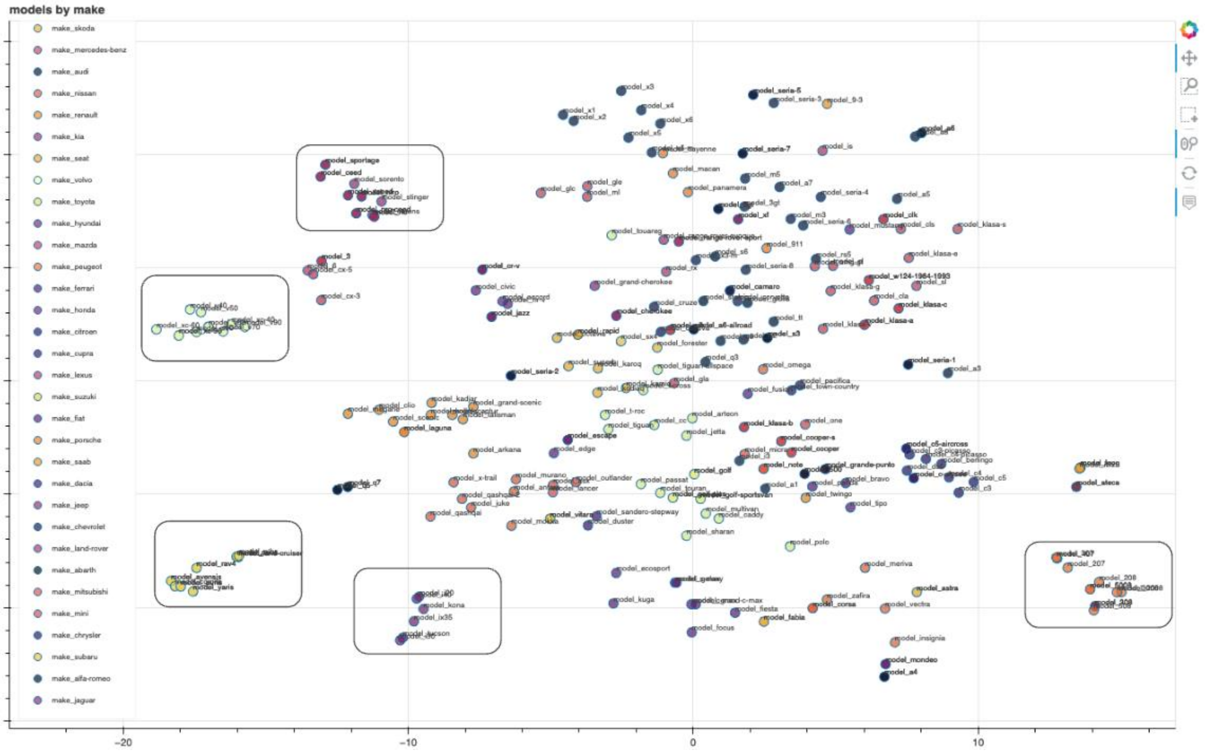


Figure 11 – Word2vec scatterplot of models by make

For example, the small clusters rectangular clusters that were deliberately drawn, are specific brands clusters. This means that users that look at those car makes have a high probability to look for other cars of the same brand, instead of switching to a different brand. Another conclusion that could be taken is that users that look at these brands have high loyalty to them. Some examples are Volvo, Toyota and Hunday. It is noticeable in the rest of the plot always a distinction of makes but a higher closeness between brands: users that look at those cars probably are more attracted by different characteristics of the car instead of their make. In the blue circle cluster instead, it is possible to notice all the cars with high-end brands, namely Mercedes-Benz, Audi, BMW, Alfa Romeo and Porshe. While in the lower circle, the orange one, the focus shifts to more low-end brands like Ford, Citroen, Opel and Fiat.

It is clear already that users search for cars based on which are their needs, and probably also their budgets. To validate this, the next scatterplot shows, for the same car models, which is the range of prices they fall into. In order to do this, the average price for every car model was considered and the prices were binned into five based on its quantile distribution. This resulted in following classes:

verylow_price	**low_price**	**medium_price**	**high_price**	**above_price**
0€ < 4350€	4350€ <= 7880€	7880€ <= 13570€	13570€ <= 26500€	>= 26500€

Table 8 – Price ranges defined for the word2vec analysis

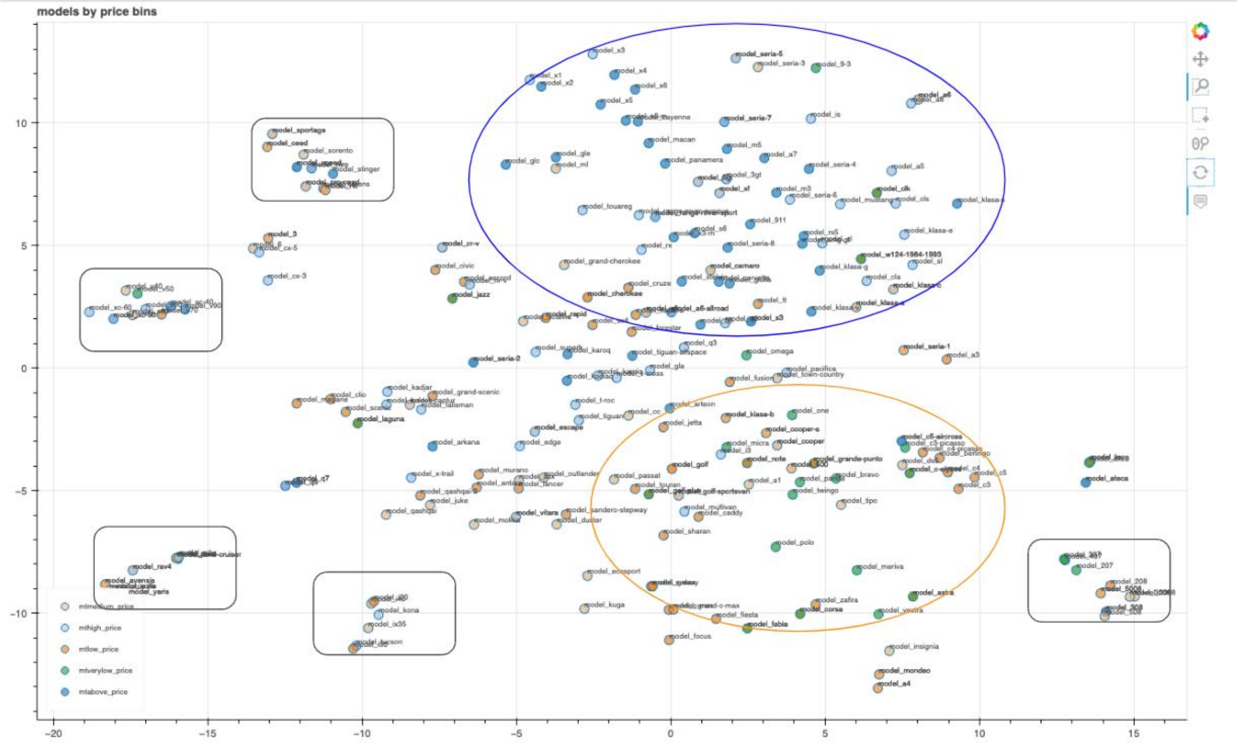


Figure 12 - Word2vec scatterplot of models by price ranges

As conjectured before, the blue cluster circle mainly covers brands which prices are higher than the rest of the cars. In fact, most of these car models belong to the high and above price ranges; while the models in the orange circle can be associated with very low, low and medium prices. Interestingly, the cars in the loyal brands are all from different price ranges.

Lastly, also the body type of the car was validated. It can be seen in the next plot.

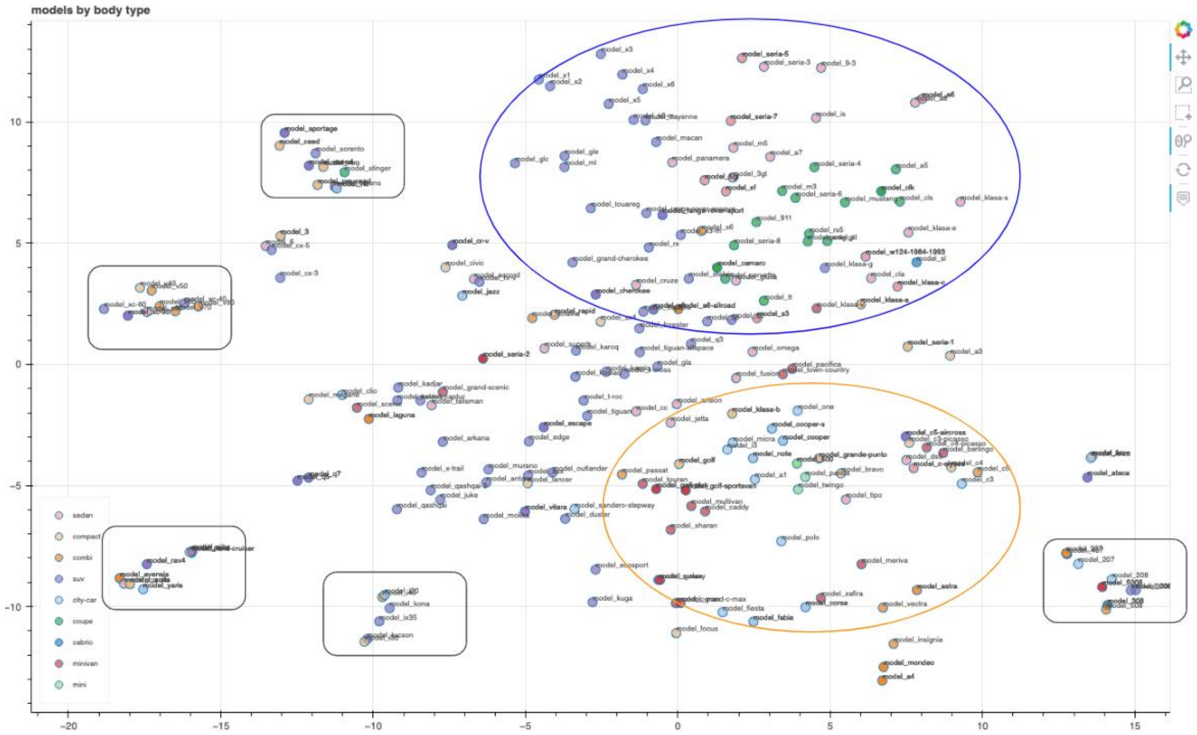


Figure 13 - Word2vec scatterplot of models by body type

It looks like not only users take their decisions based on the price range or the brand but by also the type of car, in this case the body type of it. High end cars are mainly coupes, SUVs and sedans. While in the low-end cluster, mostly minivans, city cars and compacts can be found. This ties with the initial research done by the company: users look and buy cars based on which point of their life they are in. It depends if they are starting a family, or they got a new job, or they plan to commute just in the surroundings of the city.

When computing k-means on the 25-dimensional vectors, 3 clusters are distinguishable. The scatter plot is the following:

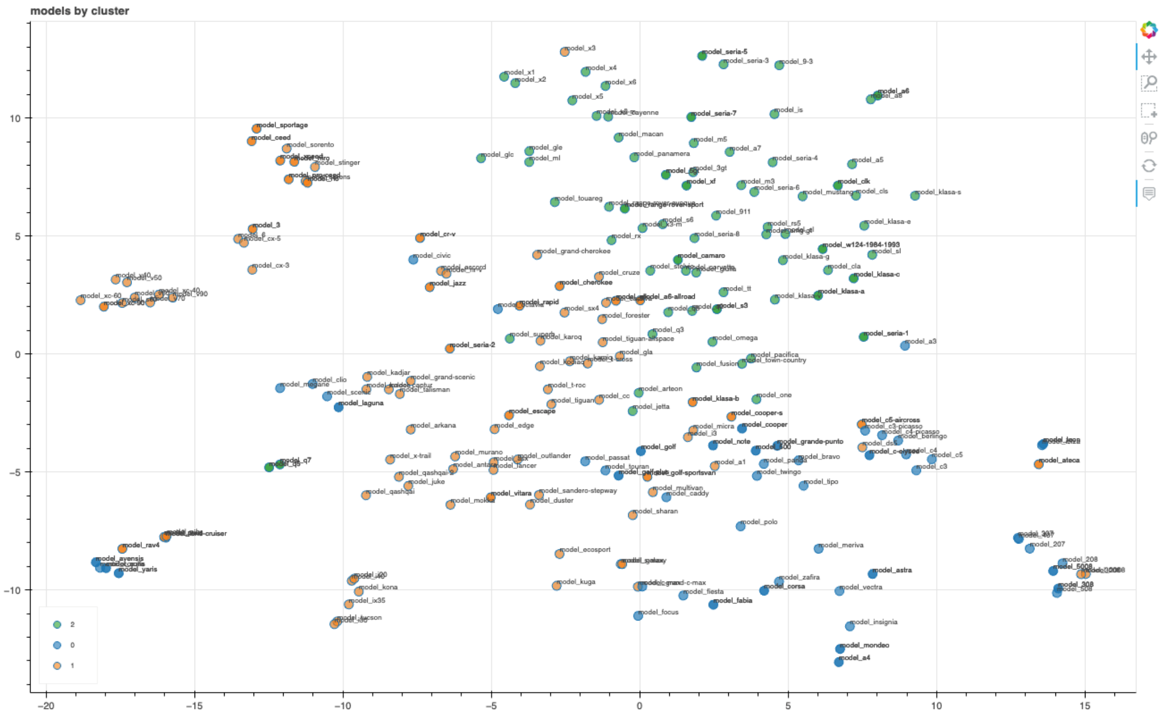


Figure 14 - Word2vec scatterplot of models by k-mean cluster

Each cluster has respectively 72, 120 and 86 car models. The cluster 0 mostly contains cars with very low, low and medium prices, while cluster 2 has a preponderance of expensive vehicles. Cluster 1 has a distribution more focused of medium and high prices.

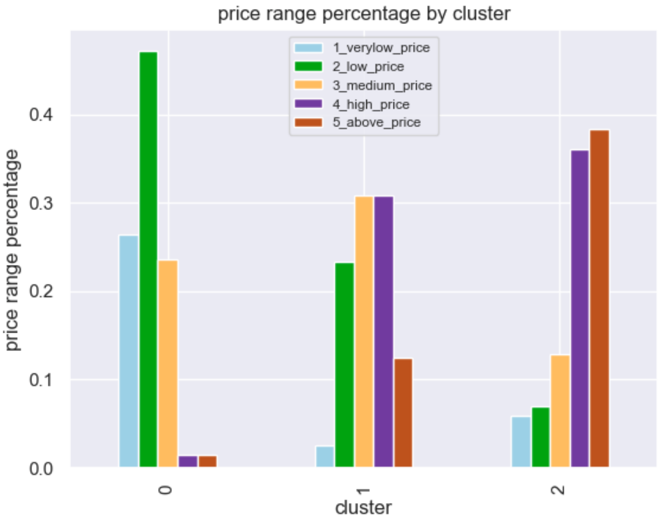


Figure 15 – K-means cluster and relative frequency of price ranges

Same analysis can be made for the body type: in the cluster 0, with low range prices, most of the vehicles are city cars, combis or compacts, while for cluster one there is a main prevalence of SUVs. These ones can also be found in cluster 2 with also 38% of the cars being sedans.

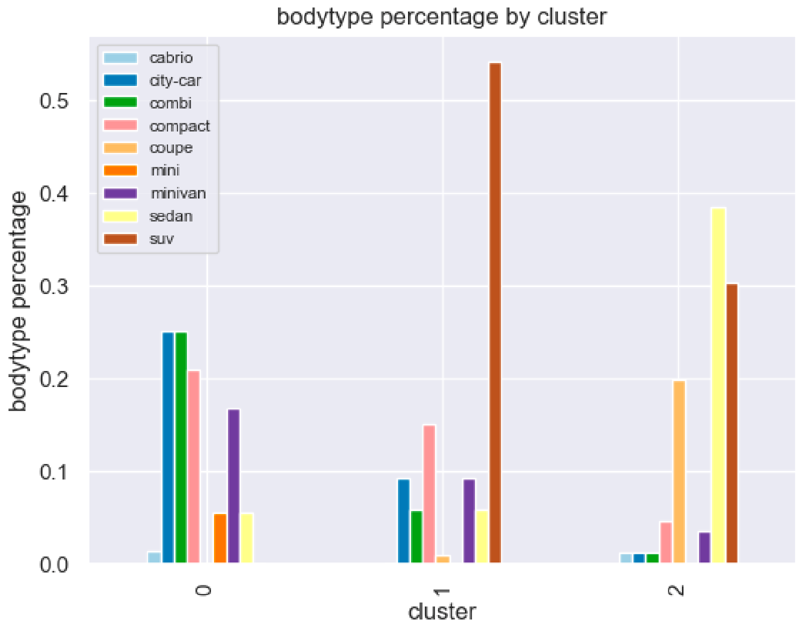


Figure 16 – K-means cluster and relative frequency of body types

4. CONCLUSIONS

The main question raised by the business was to make possible the suggestion of filters via data science techniques, but throughout one year of testing and analysis it was clear that user behaviour is an important part of our business, and it can help greatly to understand the way projects can evolve. The example in this thesis is one: from a filter suggester purpose the focus shifted to a car suggester purpose. As described until now, users tend to look at cars mostly in three lights depending on which period of their lives they are living: brand loyalty, price ranges and type of car.

Firstly, the data exploration and time analysis helped to deep dive into how users utilize our platform: which filters do they select, how many do they select in a certain search and how long does it take for them to reach a meaningful interaction.

In a second moment, after running the first AB test, a more in-depth analysis was conducted, stating that users with the most meaningful interactions and replies are the ones that see different brands and car models. Evidently this was a spark for the shift to car suggester, as after the word2vec analysis it was clear that users have different needs and need different recommendations based on which car models they are already interacting with.

The word2vec model has been later trained and used by the team to suggest similar cars of the listing page, and currently, in September 2022, it is under testing.

5. LIMITATIONS AND RECOMMENDATIONS FOR FUTURE WORKS

As for the future work, there are numerous improvements that can be tested: from utilizing word2vec results to create new recommendations, deep dive in the different types of buyers our platforms to give a more personalized and tailored search experience to every user. The goal, if the AB test is successful, is to roll out the model to the other countries in which the company operates in, namely Romania and Portugal, across both mobile and desktop devices; eventually also apps would be tackled. Other features could be included in the word2vec training dataset. For example, the amount of time users spend interacting with a specific listing, or the type of meaningful interaction they are performing. These could be added like in (Zhou et al., 2018) where they built a RNN layer on top of the embeddings one.

REFERENCES

Barkan, O., & Koenigstein, N. (2016a). ITEM2VEC: Neural item embedding for collaborative filtering. In *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*.

Barkan, O., & Koenigstein, N. (2016b). ITEM2VEC: Neural item embedding for collaborative filtering. *IEEE International Workshop on Machine Learning for Signal Processing, MLSP*, *2016-November*. https://doi.org/10.1109/MLSP.2016.7738886

ben Ellencweig, S. E. D. F. and I. M. (2019, June). *Used cars, new platforms: Accelerating sales in a digitally disrupted market*. Https://Www.Mckinsey.Com/Industries/Automotive-and-Assembly/Our-Insights/Used-Cars-New-Platforms-Accelerating-Sales-in-a-Digitally-Disrupted-Market.

Cho, H., & Yoon, S. M. (2017). Issues in Visualizing Intercultural Dialogue Using Word2Vec and t-SNE. *2017 International Conference on Culture and Computing (Culture and Computing)*, 149–150.

Chow, A., Nicole, M.-H., & Manai, G. (2014). *HybridRank : A Hybrid Content-Based Approach To Mobile Game Recommendations*.

Eric Rosenbaum. (2020, October 15). *The used car boom is one of the hottest, and trickiest, coronavirus markets for consumers*. Https://Www.Cnbc.Com/2020/10/15/Used-Car-Boom-Is-One-of-Hottest-Coronavirus-Markets-for-Consumers.Html.

Esmeli, R., Bader-El-Den, M., & Abdullahi, H. (2020). Using Word2Vec Recommendation for Improved Purchase Prediction. In *2020 International Joint Conference on Neural Networks (IJCNN)*.

Larry Hardesty. (2019, November 29). *the-history-of-amazons-recommendation-algorithm*. https://www.amazon.science/the-history-of-amazons-recommendation-algorithm

Linden, G., Smith, B., & York, J. (2003). Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, *7*(1), 76–80. https://doi.org/10.1109/MIC.2003.1167344

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013a). *Distributed Representations of Words and Phrases and their Compositionality*.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013b). *Efficient Estimation of Word Representations in Vector Space*.

Olivier Hanoulle. (2021). *The online boom in used-car sales*.

Sándor Apáthy. (2021, January 5). *History of recommender systems*.

van der Maaten, L., & Hinton, G. (2008). Visualizing Data using t-SNE. In *Journal of Machine Learning Research* (Vol. 9).

Wang, D., Liang, Y., Xu, D., Feng, X., & Guan, R. (2018). A content-based recommender system for computer science publications. *Knowledge-Based Systems*, *157*, 1–9. https://doi.org/10.1016/j.knosys.2018.05.001

Zhou, M., Ding, Z., Tang, J., & Yin, D. (2018). Micro behaviors: A new perspective in E-commerce recommender systems. *WSDM 2018 - Proceedings of the 11th ACM International Conference on Web Search and Data Mining*, *2018-Febuary*, 727–735. https://doi.org/10.1145/3159652.3159671

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação

Universidade Nova de Lisboa